



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/790,663	03/01/2004	Izydor Gryko	MSFT-2767/305783.01	9551
41505	7590	06/05/2007		
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION) CIRA CENTRE, 12TH FLOOR 2929 ARCH STREET PHILADELPHIA, PA 19104-2891			EXAMINER WANG, BEN C	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 06/05/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/790,663

Applicant(s)

GRYKO ET AL.

Examiner

Ben C. Wang

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 March 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>02/27/2006</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-21 are pending in this application and presented for examination.

Claim Rejections – 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 8-14 and 15-21 are rejected under 35 U.S.C 101 because the claims are directed to non-statutory subject matter.
3. In claim 8, a "system for building an extensible project comprising: a base project object...; ... one flavor object...; an aggregator ..." is being cited, lines 1-5; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as computer software per se, are not physical "things". They are neither computer components nor statutory processes, as they are not "act" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. Accordingly, it is important to distinguish claims

Art Unit: 2192

that define descriptive material per se from claims that define statutory inventions. (See MPEP 2106.01(I))

4. **As to claims 9-14**, they are merely further recited as the computer software per se, thus, do not cure the deficiency of base claim 8, and also rejected under 35 U.S.C. 101 as set forth above.

5. **In claim 15**, the "computer-readable medium storing instructions" is being cited, line 2, to include transmission media, light waves, a carrier wave etc., cited in [0023], lines 12-20 in the specifications; the claim is directed to a computer program product encoding a computer program. However, Applicant defines "computer-readable medium" to include "a computer data signal embodied in a carrier wave". Signals and carrier waves do not fall within any class of statutory subject matter, and thus the claim is not limited to statutory subject matter. Please see Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (1300 OG 142), Annex IV, Section (C) for details.

6. **As to claims 16-21**, they do not cure the deficiency of base claim 15, and also are rejected under 35 U.S.C. 101 as set forth above.

Claim Interpretation of Record for Claims 15-21

7. In the interest of compact prosecution, the Examiner subsequently interprets the above phrase "computer-readable medium" as being "the computer

storage medium (emphasis added) and NOT the communication media (emphasis added) (non-statutory)" for the purpose of further examination.

Specification Objections

8. The specification is objected to because the following informalities:
- "IA-64", cited in [0003], Lines 7-8, 11-13, is a registered trademark
 - "'MICROSOFT VISUAL STUDIO .NET" and "Microsoft", cited in [0006], Lines 2, 8; [0029], Lines 3, 6 respectively, are registered trademarks
 - "to the interface can be passed in to the base project object 220. In this way, the flavor can enhance an interface implemented in the base project object 220 without needing to", cited in [0032], Lines 14-15, should be corrected as "to the interface can be passed in to the base project object 200. In this way, the flavor can enhance an interface implemented in the base project object 200 without needing to"
 - "As shown in Figure 4, an IVsFlavorCfgProvider interface 430 is provided", cited in [0038], Lines 2-3, should be corrected as "As shown in Figure 4, an IVsFlavorCfgProvider interface 430(a) is provided"
 - "project configuration object, the IVsFlavorCfgProvider interface 430 is used to provided", cited in [0039], Line 2, should be corrected as "project configuration object, the IVsFlavorCfgProvider interface 430(a) is used to provided"

Art Unit: 2192

- “The IVsFlavorCfgProvider interface 430 functions”, cited in [0039], Line 5, should be corrected as “The IVsFlavorCfgProvider interface 430(a) functions”

Appropriate correction is required (See MPEP § 608.01(b))

Claim Rejections – 35 USC § 102(b)

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(b) that form the basis for the rejections under this section made in this office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-5, 8-12, and 15-19 are rejected under 35 U.S.C. 102(b) as being anticipated by Leach et al. (Pat. No. US 6,412,020 B1) (hereinafter ‘Leach’)

10. **As to claim 1**, Leach discloses a method for building an extensible project system (Abstract, Lines 1-4 – the method aggregates an enclosed object within an enclosing object) comprising: providing a base project object comprising data for creating a project system (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to client of the enclosing object; Col. 9, Lines 27-30 – to provide a method and system for enclosing objects wherein an enclosed object can itself be an enclosing object to an arbitrary level of enclosing; Col.9, Lines 45-46 – implementing controlling behavior over common functionality present in enclosed objects; Col. 10, Lines 9-

Art Unit: 2192

13 – an enclosed object is implement with knowledge of the external interfaces of the enclosed object and has no knowledge of interfaces (other than the controlling object management interface; Col. 10, Lines 35-38 – during creation, a pointer to the enclosing multi-type object is passed to the object to be enclosed to enable the enclosed object to communicate with the enclosing multi-type object) of the enclosing object or other enclosed objects); providing at least one flavor object comprising data for modifying said project system for a specific purpose (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to a client of the enclosing object; Col. 9, Lines 13-14 – enclosing an object within another object after the enclosing object is instantiated; Col. 9, Lines 39-42 – supplying default functionality to objects by enclosing them within an enclosing object where an enclosed or enclosing object implements the default functionality; Col. 9, Lines 55-58 – the enclosed object has an object management interface and on or more external interfaces, while the enclosing object has a controlling object management interface); creating a flavored project system adapted for said specific purpose by object aggregation using said base project object as a participating object and one of said at least one flavor objects as a controlling object (Col. 8, Lines 66-67 – a method and system for aggregating objects; Col. 9, Lines 4-7 – dynamically aggregating objects; statically aggregating objects; Col. 9, Lines 25-26 – implementing an aggregate object so that a client is unaware that the object is an aggregate; Col. 9, Lines 50-61 – the method aggregates an enclosed object within an enclosing object; each interface exposed to a client by the aggregate

Art Unit: 2192

object has a query function member for receiving an identifier of an interface and for returning a reference to the identified interface; Col. 10, Lines 8-13 (static aggregation), 17-24 (dynamic aggregation); Col. 10, Lines 24-30 – the multi-type object has an add interface function member, which can be used to aggregate interfaces by adding them to the enclosing multi-type object; the multi-type object also has an add object function member for aggregating all of the interface of an object; Col. 10, 34-47 – a preferred method invokes the add interface function member or the add object function member of the enclosing multi-type object passing it a reference to the created object implementing the interface to be aggregated; the query function member of the enclosing multi-type object is invoked in order to retrieve a reference to the interface that has been aggregated).

11. **As to claim 8**, Leach discloses a system for building an extensible project system (Abstract, Lines 1-4 – the method aggregates an enclosed object within an enclosing object) comprising: a base project object comprising data for creating a project system (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to client of the enclosing object; Col. 9, Lines 27-30 – to provide a method and system for enclosing objects wherein an enclosed object can itself be an enclosing object to an arbitrary level of enclosing; Col.9, Lines 45-46 – implementing controlling behavior over common functionality present in enclosed objects; Col. 10, Lines 9-13 – an enclosed object is implement with knowledge of the external interfaces of

Art Unit: 2192

the enclosed object and has no knowledge of interfaces (other than the controlling object management interface; Col. 10, Lines 35-38 – during creation, a pointer to the enclosing multi-type object is passed to the object to be enclosed to enable the enclosed object to communicate with the enclosing multi-type object) of the enclosing object or other enclosed objects); at least one flavor object comprising data for modifying said project system for a specific purpose (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to a client of the enclosing object; Col. 9, Lines 13-14 – enclosing an object within another object after the enclosing object is instantiated; Col. 9, Lines 39-42 – supplying default functionality to objects by enclosing them within an enclosing object where an enclosed or enclosing object implements the default functionality; Col. 9, Lines 55-58 – the enclosed object has an object management interface and on or more external interfaces, while the enclosing object has a controlling object management interface); an aggregator for creating a flavored project system for said specific purpose by object aggregation using said base project object as a participating object and one of said at least one flavor objects as a controlling object (Col. 8, Lines 66-67 – a method and system for aggregating objects; Col. 9, Lines 4-7 – dynamically aggregating objects; statically aggregating objects; Col. 9, Lines 25-26 – implementing an aggregate object so that a client is unaware that the object is an aggregate; Col. 9, Lines 50-61 – the method aggregates an enclosed object within an enclosing object; each interface exposed to a client by the aggregate object has a query function member for receiving an identifier of an interface and

for returning a reference to the identified interface; Col. 10, Lines 8-13 (static aggregation), 17-24 (dynamic aggregation); Col. 10, Lines 24-30 – the multi-type object has an add interface function member, which can be used to aggregate interfaces by adding them to the enclosing multi-type object; the multi-type object also has an add object function member for aggregating all of the interface of an object; Col. 10, 34-47 – a preferred method invokes the add interface function member or the add object function member of the enclosing multi-type object passing it a reference to the created object implementing the interface to be aggregated; the query function member of the enclosing multi-type object is invoked in order to retrieve a reference to the interface that has been aggregated).

12. **As to claim 15**, Leach discloses a computer-readable medium for building an extensible project system (Abstract, Lines 1-4 – the method aggregates an enclosed object within an enclosing object), said computer readable-medium storing instructions for causing a computer to perform the steps of comprising: providing a base project object comprising data for creating a project system (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to client of the enclosing object; Col. 9, Lines 27-30 – to provide a method and system for enclosing objects wherein an enclosed object can itself be an enclosing object to an arbitrary level of enclosing; Col.9, Lines 45-46 – implementing controlling behavior over common functionality present in enclosed objects; Col. 10, Lines 9-13 – an enclosed object is

Art Unit: 2192

implement with knowledge of the external interfaces of the enclosed object and has no knowledge of interfaces (other than the controlling object management interface; Col. 10, Lines 35-38 – during creation, a pointer to the enclosing multi-type object is passed to the object to be enclosed to enable the enclosed object to communicate with the enclosing multi-type object) of the enclosing object or other enclosed objects); providing at least one flavor object comprising data for modifying said project system for a specific purpose (Col. 9, Lines 9-11 – enclosing an object within another object while exposing an interface of the enclosed object to a client of the enclosing object; Col. 9, Lines 13-14 – enclosing an object within another object after the enclosing object is instantiated; Col. 9, Lines 39-42 – supplying default functionality to objects by enclosing them within an enclosing object where an enclosed or enclosing object implements the default functionality; Col. 9, Lines 55-58 – the enclosed object has an object management interface and on or more external interfaces, while the enclosing object has a controlling object management interface); creating a flavored project system for said specific purpose by object aggregation using said base project object as a participating object and one of said at least one flavor objects as a controlling object (Col. 8, Lines 66-67 – a method and system for aggregating objects; Col. 9, Lines 4-7 – dynamically aggregating objects; statically aggregating objects; Col. 9, Lines 25-26 – implementing an aggregate object so that a client is unaware that the object is an aggregate; Col. 9, Lines 50-61 – the method aggregates an enclosed object within an enclosing object; each interface exposed to a client by the aggregate object has a query function

member for receiving an identifier of an interface and for returning a reference to the identified interface; Col. 10, Lines 8-13 (static aggregation), 17-24 (dynamic aggregation); Col. 10, Lines 24-30 – the multi-type object has an add interface function member, which can be used to aggregate interfaces by adding them to the enclosing multi-type object; the multi-type object also has an add object function member for aggregating all of the interface of an object; Col. 10, 34-47 – a preferred method invokes the add interface function member or the add object function member of the enclosing multi-type object passing it a reference to the created object implementing the interface to be aggregated; the query function member of the enclosing multi-type object is invoked in order to retrieve a reference to the interface that has been aggregated).

13. **As to claim 2** (incorporating the rejection in claim 1), Leach discloses the method where said at least one flavor object comprises at least a first flavor object and a second flavor object, and where said step of creating a flavored project system comprises: creating an intermediary object by aggregating said first flavor object as a controlling object and said base project object as a participating object; and creating a flavored project system by using said second flavor object as a controlling object and said intermediary object as a participating object (Col. 9, Lines 27-30 – to provide a method and system for enclosing objects where an enclosed object can itself be an enclosing object to an arbitrary level of enclosing).

14. **As to claim 3** (incorporating the rejection in claim 1), Leach discloses the method where said step of creating a flavored project system comprises allowing at least one interface of said base project to be modified by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

15. **As to claim 4** (incorporating the rejection in claim 3), Leach discloses the method where said step of creating a flavored project system comprises allowing a value for at least one property stored in said at least one interface of said base project to be modified by a value for said at least one property stored in an interface of said flavor object (Col. 5, Lines 10-12 – the overriding virtual function can modify the state of the object in a way that affects non-overridden functions; Col. 9, Lines 1-2 – to provide a method and system for dynamically modifying object behavior; Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object; Col. 10, Lines 17-24 – an object can be modified dynamically by allowing interface

Art Unit: 2192

instances, as implemented by objects, to be aggregated during the execution of a client program).

16. **As to claim 5** (incorporating the rejection in claim 1), Leach discloses the method where said step of creating a flavored project system comprises allowing at least one interface of said base project to be replaced by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

17. **As to claim 9** (incorporating the rejection in claim 8), Leach discloses the system where said at least one flavor object comprises at least a first flavor object and a second flavor object, and where said aggregator further comprises: a first aggregator for creating an intermediary object by aggregating said first flavor object as a controlling object and said base project object as a participating object; and a second aggregator for creating a flavored project system by using said second flavor object as a controlling object and said intermediary object as a participating object (Col. 9, Lines 27-30 – to provide a method and system for enclosing objects where an enclosed object can itself be an enclosing object to an arbitrary level of enclosing).

18. **As to claim 10** (incorporating the rejection in claim 8), Leach discloses the system where said aggregator causes at least one interface of said base project to be modified by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

19. **As to claim 11** (incorporating the rejection in claim 10), Leach discloses the system where said aggregator causes a value for at least one property stored in said at least one interface of said base project to be modified by a value for said at least one property stored in an interface of said flavor object (Col. 5, Lines 10-12 – the overriding virtual function can modify the state of the object in a way that affects non-overridden functions; Col. 9, Lines 1-2 – to provide a method and system for dynamically modifying object behavior; Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object; Col. 10, Lines 17-24 – an object can be modified dynamically by allowing interface instances, as implemented by objects, to be aggregated during the execution of a client program).

20. **As to claim 12** (incorporating the rejection in claim 8), Leach discloses the system where said aggregator causes at least one interface of said base project to be replaced by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

21. **As to claim 16** (incorporating the rejection in claim 15), Leach discloses the computer-readable medium where said at least one flavor object comprises at least a first flavor object and a second flavor object, and where said step of creating a flavored project system comprises: creating an intermediary object by aggregating said first flavor object as a controlling object and said base project object as a participating object; and creating a flavored project system by using said second flavor object as a controlling object and said intermediary object as a participating object (Col. 9, Lines 27-30 – to provide a method and system for enclosing objects where an enclosed object can itself be an enclosing object to an arbitrary level of enclosing).

22. **As to claim 17** (incorporating the rejection in claim 15), Leach discloses the computer-readable medium where said step of creating a flavored project

Art Unit: 2192

system comprises allowing at least one interface of said base project to be modified by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

23. **As to claim 18** (incorporating the rejection in claim 17), Leach discloses the computer-readable medium where said step of creating a flavored project system comprises allowing a value for at least one property stored in said at least one interface of said base project to be modified by a value for said at least one property stored in an interface of said flavor object (Col. 5, Lines 10-12 – the overriding virtual function can modify the state of the object in a way that affects non-overridden functions; Col. 9, Lines 1-2 – to provide a method and system for dynamically modifying object behavior; Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object; Col. 10, Lines 17-24 – an object can be modified dynamically by allowing interface instances, as implemented by objects, to be aggregated during the execution of a client program).

24. **As to claim 19** (incorporating the rejection in claim 15), Leach discloses the computer-readable medium where said step of creating a flavored project system comprises allowing at least one interface of said base project to be replaced by said flavor object (Col. 9, Lines 34-37 – for enhancing a base object's apparent behavior by adding an interface to it that overrides standard behavior of the base object; Col. 25, Lines 3-16 – these combining rules can be used to override the standard behavior of an enclosed base object by providing access to a new implementation of a previously defined interface of the enclosed base object).

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

25. **Claims 6-7, 13-14, and 20-21** are rejected under 35 U.S.C. 103(a) as being unpatentable over Leach in view of Williams et al., (Pat. No. US 6,256,780 B1) (hereinafter 'Williams')

26. **As to claim 6** (incorporating the rejection in claim 1), Leach does not explicitly disclose the method said creating a flavored project system comprising:

providing a first delegate object associated with said base project object; and providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object.

However, in an analogous art of method and system for assembling software components, Williams discloses the method said creating a flavored project system comprising: providing a first delegate object associated with said base project object; and providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object (Col. 4, Lines 7-43; Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index “i1”, represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role “r1”, represented by socket 103b; Col. 8, Lines 29-48; Col. 9, Lines 42-53; Col. 10, Lines 27-32).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Williams into the Leach’s system to further provide a first delegate object associated with said base project object; and providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object in Leach’s system.

The motivation is that it would further enhance the Leach's system by taking, advancing and/or incorporating Williams's system which offers significant advantages for providing a light-weight mechanism for instantiating the component objects of an assembly object and a light-weight mechanism for controlling the exporting and importing of objects; thus, the assembly system relieves the developer of an assembly object from the task of developing code for such instantiation and control as once suggested by Williams (Col. 2, Lines 37-44).

27. **As to claim 7** (incorporating the rejection in claim 6), Williams discloses the method where said second delegate object includes an extender interface, said creation of a project system further comprising: providing an extender site object associated with said extender interface (Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index "i1", represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role "r1", represented by socket 103b).

28. **As to claim 13** (incorporating the rejection in claim 8), Leach does not explicitly disclose the system said project system comprising: a first delegate object associated with said base project object; and a second delegate object

Art Unit: 2192

associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object.

However, in an analogous art of method and system for assembling software components, Williams discloses the system said project system comprising: a first delegate object associated with said base project object; and a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object (Col. 4, Lines 7-43; Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index “i1”, represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role “r1”, represented by socket 103b; Col. 8, Lines 29-48; Col. 9, Lines 42-53; Col. 10, Lines 27-32).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Williams into the Leach’s system to further provide the system said project system comprising: a first delegate object associated with said base project object; and a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object in Leach’s system.

The motivation is that it would further enhance the Leach's system by taking, advancing and/or incorporating Williams's system which offers significant advantages for providing a light-weight mechanism for instantiating the component objects of an assembly object and a light-weight mechanism for controlling the exporting and importing of objects; thus, the assembly system relieves the developer of an assembly object from the task of developing code for such instantiation and control as once suggested by Williams (Col. 2, Lines 37-44).

29. **As to claim 14** (incorporating the rejection in claim 13), Williams discloses the system where said second delegate object includes an extender interface, said project system further comprising: an extender site object associated with said extender interface (Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index "i1", represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role "r1", represented by socket 103b).

30. **As to claim 20** (incorporating the rejection in claim 15), Leach does not explicitly disclose the computer-readable medium said creating a project system by object aggregation using said base project object as a participating object and

Art Unit: 2192

one of said at least one flavor objects as a controlling object comprising:
providing a first delegate object associated with said base project object; and
providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object.

However, in an analogous art of method and system for assembling software components, Williams discloses the computer-readable medium said creating a project system by object aggregation using said base project object as a participating object and one of said at least one flavor objects as a controlling object comprising: providing a first delegate object associated with said base project object; and providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object (Col. 4, Lines 7-43; Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index “i1”, represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role “r1”, represented by socket 103b; Col. 8, Lines 29-48; Col. 9, Lines 42-53; Col. 10, Lines 27-32).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Williams into the Leach’s system to further provide the computer-readable medium said creating a

project system by object aggregation using said base project object as a participating object and one of said at least one flavor objects as a controlling object comprising: providing a first delegate object associated with said base project object; and providing a second delegate object associated with one of said at least one flavor objects, where said second delegate object extends the functionality of said first delegate object in Leach's system.

The motivation is that it would further enhance the Leach's system by taking, advancing and/or incorporating Williams's system which offers significant advantages for providing a light-weight mechanism for instantiating the component objects of an assembly object and a light-weight mechanism for controlling the exporting and importing of objects; thus, the assembly system relieves the developer of an assembly object from the task of developing code for such instantiation and control as once suggested by Williams (Col. 2, Lines 37-44).

31. **As to claim 21** (incorporating the rejection in claim 20), Williams discloses the computer-readable medium where said second delegate object includes an extender interface, said creation of a project system further comprising: providing an extender site object associated with said extender interface (Fig. 1 – illustrating an assembly object along with its connection to external entities; Col. 4, Line 52 through Col. 5, Line 20 – an external entity connects assembly-2 to assembly-3 by retrieving the reference to a connector of assembly-2 and requesting the connector to export the element identified by index "i1",

Art Unit: 2192

represented by plug102a. the external entity then requests connector-3 of assembly-3 to connect assembly-2 through the connection identified by role "r1", represented by socket 103b).

Conclusion

32. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


- Leach et al., Method and System for Aggregating Objects (Pat. No. US 5,805,885)
- Leach et al., Method and System for Aggregating Objects (Pat. No. US 5,710,925)

33. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TUAN DAM
SUPERVISORY PATENT EXAMINER

BCW *FW*

May 24, 2007